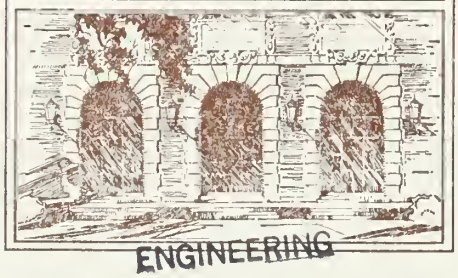


Center for Advanced Computation

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

URBANA ILLINOIS 61801

CAC Document No. 12

THE QR-ALGORITHM

by

Masako Ogura

September 1, 1971

CAC Document No. 12

THE QR-ALGORITHM

by

Masako Ogura

Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

September 1, 1971

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by the U.S. Army Research Office - Durham under Contract No. DAHC04-72-C-0001.

ABSTRACT

The implementation of QR-algorithm on ILLIAC IV is described. An ASK subroutine for computing all eigenvalues of a real Hessenberg matrix of order less than or equal to 64 by this algorithm is attached. The QR-transformation consists of the decomposition of the matrix A_k into the product of a unitary matrix Q_k and an upper triangular matrix R_k, and forming A_{k+1} by post-multiplying R_k by Q_k, where $A_1 = A$ is the original matrix. All eigenvalues are either isolated on the diagonal or are eigenvalues of a 2 x 2 diagonal submatrix as $k \to \infty$.

TABLE OF CONTENTS

1. INTRODUCTION

An ASK program for finding all eigenvalues of a real Hessenberg matrix of order less than or equal to 64 is written and tested on the B5500 simulator. The QR-algorithm for real Hessenberg matrices described herein is that of Martin et al [2]. The ILLIAC IV computer time required for performing one iteration (computation of A_{k+2} from A_k, refer to 4.1) on a 64 x 64 matrix is approximately 10 millisecond.

The necessary information for using this program is given in Section 2. The test result of this program on a 4 x 4 real Hessenberg matrix is given in Section 3. In Section 4, the outline of the QR-algorithm is given and Section 5 is devoted to the actual programming technique to implement this algorithm on the ILLIAC IV computer. The flow chart and ASK program are attached as Appendices 1 and 2 respectively.

2. USAGE

This program assumes that the given real Hessenberg matrix is stored in the core memory in the straight storage scheme so that each row is stored across the PE's, starting with PE0. The real and imaginary parts of the eigenvalues found are to be stored in the two rows in the PE memory specified by the user. The original matrix is destroyed and replaced by the matrix which results from the QR transformations. The content of ACAR2 and ACAR3 are destroyed since the ACAR3 is used for linkage between the subroutine and main program and ACAR2 is used for passing the address of parameters to the subroutine.

2.1 Calling sequence

Calling sequence for this subroutine is:

CALL HQR (N, A, WR, WI, IT)

where A designates the first row of the matrix and is declared in the main program as

A: DATA $a_{00}, a_{01}, \ldots, a_{0,N-1}, (0.0)M,$

$a_{10}, a_{11}, \ldots, a_{1,N-1}, (0.0)M,$

. . . .

. . . .

$a_{N-1,0}, a_{N-1,1}, \ldots, a_{N-1,N-1}, (0.0)M;$

where M = 64 - N.

N is the size of the matrix declared as

N: EQU 64;

or

DEFINE N = 64 ##;

or given as an integer, i.e., 64.

WR and WI are the rows in the PE memory where resulting real and imaginary parts of the eigenvalues respectively are to be stored. They are declared as

WR: BLK 1; and WI: BLK 1;

IT is the row vector in memory to which the number of iterations required for finding each eigenvalue is to be placed. This is declared as

IT: BLK 1;

If zero is placed in place of IT, it is to be considered that a user does not want to know the number of iterations required. The CALL macro should be defined in the user's program as:

```
DEFINE CALL  &NAME (&PARAMETERS) =
        &IF  &SIGN (&MFIELD(&NAME))
             &THEN EXTERNAL &NAME; &FI
        &IF  &EMPTY (&PARAMETERS) &THEN &ELSE
        BEGIN BLOCK
             BEGIN USE (63)
             LIST:  DATA &PARAMETERS
             END;
             CLC(2);
             SLIT(2) LIST;
        END; &FI
        CLC(3);
        SLIT(3) &NAME;
        EXCHL(3) &ICR ##;
```

2.2 Core storage used

This routine uses 500 words of PE memory for storing instructions. One row is used for storing PE numbers and three additional rows are used for temporary storage. ADBO ~ 32 are also used.

2.3 Constant

EPS (ϵ), the constant which is used to test the convergence (4.1), is taken as 10^{-10} in this program. If a user wants to change the value of this constant, he may insert EPS: DATA (desirable value); in place of

```
EPS:  DATA @ - 10; .
```

3. EXAMPLE

A test of this program was made on the B5500 simulator for the matrix:

$$\begin{bmatrix} 5.0 & -2.0 & -5.0 & -1.0 \\ 1.0 & 0.0 & -3.0 & 2.0 \\ 0.0 & 2.0 & 2.0 & -3.0 \\ 0.0 & 0.0 & 1.0 & -2.0 \end{bmatrix}$$

with $\epsilon = 10^{-8}$. The comparison of the eigenvalues obtained by this program to the exact values is given in Table 1. The selection of this small matrix and a relatively large ϵ was made because of the speed of the SSK simulator on the B5500; the execution speed ratio of the simulator to the ILLIAC IV is approximately $1:10^6$.

Table 1

Eignevalues obtained on B5500 simulator	Exact eigenvalues
3.999999997867	4.0
1.000000001066 + 1.999999999928i	1.0 + 2.0i
1.000000001066 - 1.999999999928i	1.0 - 2.0i
-1.000000000000	-1.0

4. QR-ALGORITHM

4.1 Brief outline of the QR-algorithm

The QR-transformation consists of the decomposition of the matrix A_k into the product of a unitary matrix Q_k and an upper triangular matrix R_k, and forming A_{k+1} by post-multiplying R_k by Q_k. Thus

$$A_{k+1} = R_k Q_k \qquad \text{where} \qquad A_k = Q_k R_k, \tag{1}$$

therefore

$$A_{k+1} = Q_k^* A_k Q_k$$

where $A_1 = A$ is the original matrix. It can be shown in general that A_k tends to a form in which $a^{(k)}_{i+1,\,i}\, a^{(k)}_{i+2,\,i+1} = 0$ for $i = 0, 1, \ldots, N-3$ as k increases. All eigenvalues are therefore either isolated on the diagonal or they are eigenvalues of a 2 x 2 diagonal submatrix. The amount of calculations involved in a QR step is greatly reduced if the matrix A is in the Hessenberg (or almost triangular) form. Since there are several stable methods available to reduce a general matrix to this form (ASK program HSBG is written for this purpose), the QR-algorithm is used after such reduction.

In order to achieve rapid convergence, it is essential that the origin shifts be applied and that each shift be close to an eigenvalue of the matrix. The QR-algorithm with shift of an origin s_k is expressed as:

$$A_{k+1} = R_k Q_k + s_k I \qquad \text{where} \qquad A_k - s_k I = Q_k R_k \tag{2}$$

or in other words

$$A_{k+1} = Q_k^* A_k Q_k.$$

However, even when A_1 is real, some of the eigenvalues may be complex. If the transformation (2) is carried out with a complex value of s_k, A_{k+1} is in general a complex matrix. This deficiency can be overcome by performing two steps of (2) with shifts of s_k and s_{k+1} respectively. Since s_k and s_{k+1} are both real or complex conjugate in this transformation, A_{k+2} should be always real. This transformation is described as

$$A_{k+2} = Q_{k+1}^{*} Q_k^{*} A_k Q_k Q_{k+1}$$

and
$$(Q_k Q_{k+1})(R_{k+1} R_k) = (A_k - s_k I)(A_k - s_{k+1} I) \qquad (3)$$

One method of calculating A_{k+2} by (3) would be to form the real matrix $\Gamma = (A_k - s_k I)(A_k - s_{k+1} I)$, computing its unitary-triangular decompositon to obtain $Q_k Q_{k+1}$ and transform A_k by means of this, thus giving A_{k+2}. This process requires a prohibitive amount of work, but it is shown [1] that when the matrix is in the Hessenberg form, it is unnecessary to compute more than the first column of Γ, and that this immediately gives the transformation to be applied to A_k.

4.2 Practical computation

If (3) is rewritten as

$$A_{k+2} = W^{*} A_k W \qquad \text{and} \qquad W^{*}\Gamma = \Delta$$

where $W = Q_k Q_{k+1}$ and Δ is the triangular matrix $R_{k+1} R_k$, W^{*} is a unitary matrix which reduced Γ to the triangular Δ, and W is composed of N unitary factors of the form $M_i = \begin{bmatrix} I_{i-1} & 0 \\ 0 & B_i \end{bmatrix}$ so that $W = M_1 M_2 \cdots M_k$. From the form of each M_i we see that the first column of W is equal to the first column of M_1, and this is any unitary matrix, the transpose of which eliminates the elements of the first column of Γ below diagonal.

Since we wish to transform A_k to A_{k+2} by W, we first operate on A_k with M_1. This will change the first three rows and columns of A_k since the first column of Γ contains only three non-zero elements. It follows;

$$A_k = \begin{bmatrix} a_{00} & a_{01} & a_{02} \cdots a_{0,N-1} \\ a_{10} & a_{11} & a_{12} \cdots a_{1,N-1} \\ & a_{21} & a_{22} \cdots a_{2,N-1} \\ & & \\ & & \\ & & a_{N-1,N-1} \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} \underline{a'_{00}} & \underline{a'_{01}} & \underline{a'_{02}} & \underline{a_{03}} \cdots \underline{a_{0,N-1}} \\ \underline{a'_{10}} & \underline{a'_{11}} & \underline{a'_{12}} & \underline{a_{13}} \cdots \underline{a_{1,N-1}} \\ \underline{a'_{20}} & \underline{a'_{21}} & \underline{a'_{22}} & \underline{a_{23}} \cdots \underline{a_{2,N-1}} \\ a'_{30} & a'_{31} & a'_{32} & a_{33} \cdots a_{3,N-1} \\ & & & a_{43} \cdots a_{4,N-1} \\ & & & \cdots\cdots\cdots \\ & & & \cdots\cdots\cdots \\ & & & a_{N-1,N-1} \end{bmatrix}$$

where the elements changed by the row and column operations are underlined and primed respectively. The resulting matrix is no longer in the Hessenberg form and since A_{k+2} is in the Hessenberg form, we can say that the matrices $M_2 \cdot M_3 \cdot \ldots M_N$ reduce $M_1^* A_k M_1$ to the Hessenberg form. In the practical computation, therefore, it is necessary to compute only the first column of Γ.

After each iteration (we call the calculation of A_{k+2} from A_k an iteration), all subdiagonal elements of A_{k+2} are examined to see if any of them are "negligibly small." If so, the eigenproblem for the current matrix splits into that for two or more Hessenberg matrices of smaller sizes, and the iterations continue with the submatrix in the bottom right-hand corner. It may happen that while no individual subdiagonal element is sufficiently small to be regarded as negligible, the product of two consecutive elements may be small enough to permit us to work with a submatrix. Therefore the examination of the matrix A_{k+2} is performed to see if any two consecutive subdiagonal elements are small.

5. PROGRAM DESCRIPTION

5.1 Search for negligible subdiagonal elements

We assume that the size of the matrix under consideration is $(n + 1) \times (n + 1)$ where n takes integer values between 1 and N - 1. If the last negligible subdiagonal element is in position $(\ell, \ell - 1)$, it is required only to work on the submatrix in the rows and columns ℓ to n. If none of the subdiagonal elements are negligible, ℓ is taken to be 0. The following criterion is used

$$|a_{\ell, \ell-1}| \leq \epsilon (|a_{\ell-1, \ell-1}| + |a_{\ell, \ell}|).$$

This criterian examines whether $a_{\ell, \ell-1}$ is negligible compared to the local diagonal elements.

On each PE ℓ, the following computations are simultaneously performed:

$$f(\ell) \equiv |a_{\ell, \ell-1}| - \epsilon (|a_{\ell-1, \ell-1}| + |a_{\ell, \ell}|$$

$$\text{for } \ell = 1, 2, \ldots, n.$$

If $f(\ell)$ is negative, 1 is placed in ℓth bit of the ACAR. Then searching is made for the lowest bit of the ACAR which contains 1. For example, in the following case:

	0	1	2	3 . . .	19	20	. . . . .	63
ACAR	0	0	1	0 . . .	0	1	0 0 . . .	0 0

ℓ is set to be 20.

Then test is made if $\ell = n$ or $\ell = n - 1$. If $\ell = n$, one eigenvalue is found in the place (n, n) and the matrix is deflated by 1, and n is decreased by one. In the case that $\ell = n - 1$, two eigenvalues are found as the eigenvalues of the bottom-right hand corner 2 x 2 submatrix. Then two columns and rows are deflated and n is decreased by 2.

5.2 Shifts of origin

The shifts of origin at each stage are taken to be the two roots, s_k and s_{k+1}, of the 2 x 2 matrix in the bottom right-hand corner of the current A_k. This gives

$$s_k + s_{k+1} = a_{n-1,\,n-1} + a_{n,\,n}$$
$$\text{and} \qquad s_k s_{k+1} = a_{n-1,\,n-1}\, a_{n,n} - a_{n-1,\,n}\, a_{n,\,n-1} . \tag{4}$$

In some rare cases, the process fails to converge with these shifts of origin. An example of such failure is provided by matrices of the type:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ & 1 & 0 & 0 & 0 \\ & & 1 & 0 & 0 \\ & & & 1 & 0 \end{bmatrix} .$$

Here the shifts of origin, given in (4), are both zero, and since the matrix is orthogonal, it is invariant with respect to the QR transformation without shifts. However, if one iteration is performed with any shifts of origin which are loosely related to the norm of the matrix, the convergence is very rapid. Therefore in the case where ten iterations do not produce an eigenvalue, the usual shifts s_k and s_{k+1} are replaced by shifts defined by

$$s_k + s_{k+1} = 1.5\,(|a_{n,\,n-1}| + |a_{n-1,\,n-2}|) .$$
$$s_k s_{k+1} = (|a_{n,\,n-1}| + |a_{n-1,\,n-2}|)^2 \tag{5}$$

This strategy is used again after 20 unsuccessful iterations. If 30 unsuccessful iterations are needed then a failure indication is given.

In this program, ITS is the name of an ADB where the iteration count is stored. When ITS ≠ 10, 20, we form $S = s_k + s_{k+1}$ and $Y = s_k s_{k+1}$ on a PE n according to (4) and store in ADB's. When ITS = 10 or 20, scheme (5) is used for computing S and Y. If ITS = 30, it is assumed that this algorithm fails to produce eigenvalues. As a result, only eigenvalues computed prior to this point are given in WR and WI.

5.3 Search for two consecutive small subdiagonal elements

After determining ℓ, (5.1), the submatrix in the rows ℓ to n are examined to see if any two consecutive subdiagonal elements are small enough to work with an even smaller submatrix. To test if we are to start at the row m, we compute the elements p_m, q_m and r_m such that

$$\begin{aligned} p_m &= a_{mm}^2 - a_{mm}(s_k + s_{k+1}) + s_k s_{k+1} + a_{m,m+1}\, a_{m+1,m} \\ q_m &= a_{m+1,m}(a_{mm} + a_{m+1,m+1} - s_k - s_{k+1}) \\ r_m &= a_{m+2,m+1}\, a_{m+1,m}. \end{aligned} \tag{6}$$

The criterion applied is

$$\begin{aligned} |a_{m,m-1}|\,(|q_m| + |r_m|) \\ \leq \epsilon\,|p_m|\,(|a_{m+1,m+1}| + |a_{m,m}| + |a_{m-1,m-1}|) \end{aligned} \tag{7}$$

where we test whether or not the elements which appear in the positions $(m+1, m)$, $(m+2, m+1)$ are negligible compared with the three local diagonal elements $a_{m+1,m+1}$, $a_{m,m}$ and $a_{m-1,m-1}$.

Here we take m to be the largest integer ($\geq \ell$) for which condition (6) is satisfied.

For this computation, the mode bits are turned on for PE ℓ through PE n. The p_m, q_m and r_m for $m = \ell, \ell - 1, \ldots, n$ are computed according to (6) on all PE's whose mode bits are turned on, and comparison is made to see whether (7) is satisfied. If (7) is satisfied on PE m, 1 is placed in the mth bit of the ACAR. Then the search is made for the lowest bit of the ACAR which contains 1, and m is set equal to this bit number. If no 1 is found in the ACAR, m is taken to be ℓ.

5.4 Double QR-transformation

A_{k+2} is computed by applying the QR-double transformation to A_k in such a way that $A_{k+2} = N_n^* \cdot \cdot \cdot \cdot N_m^* A_k N_m \cdot \cdot \cdot \cdot N_n$

where $N_i^* = I - \frac{U_i U_i^*}{2K_i^2}$ and $U_i = (p_i \pm t_i, q_i, r_i, 0 \cdot \cdot \cdot \cdot 0)$.

Here

$$p_i = a_{ii}^2 - a_{ii}(s_k + s_{k+1}) + s_k s_{k+1} + a_{i,i+1} a_{i+1,i}$$

$$q_i = a_{i+1,i}(a_{ii} + a_{i+1,i+1} - s_k - s_{k+1})$$

$$r_i = a_{i+2,i+1} a_{i+1,i} \qquad \text{for } i = m$$

and

$$p_i = a_{i,i-1} \pm t_i, \qquad q_i = a_{i+1,i-1}$$

$$\text{and } r_i = a_{i+2,i-1} \qquad \text{for } i \neq m.$$

t_i and $2K_i^2$ are defined as

$$t_i = \pm \sqrt{p_i^2 + q_i^2 + r_i^2}$$

$$2K_i^2 = t_i^2 \mp p_i t_i.$$

Row modification:

For $i = m, m+1, \cdot \cdot \cdot, n$, the elements of $N_i^* A_k^{(i)} \equiv N_i^* (N_{i-1}^* \cdot \cdot \cdot N_m^* A_k N_m \cdot \cdot \cdot N_{i-1})$ are different from those of $A_k^{(i)}$ in only three rows, i.e., ith, (i + 1)th and (i + 2)th rows. These new elements are computed in the following way with the elements of $A_k^{(i)}$ denoted by a_{hj}:

$$(i,j)\text{ - element} = a_{ij} - [(p_i \pm t_i)a_{ij} + q_i a_{i+1,j} + r_i a_{i+2,j}] \frac{1}{t_i}$$

$$(i+1,j)\text{ - element} = a_{i+1,j} - [(p_i \pm t_i)a_{ij} + q_i a_{i+1,j} + r_i a_{i+2,j}] \frac{q_i}{2K_i^2}$$

$$(i+2,j)\text{ - element} = a_{i+2,j} - [(p_i \pm t_i)a_{ij} + q_i a_{i+1,j} + r_i a_{i+2,j}] \frac{r_i}{2K_i^2}$$

$$\text{for } j = i, i+1, \cdot \cdot \cdot, n$$

In the actual computation, t_i and $2K_i^2$ are first computed and then p_i, q_i and r_i are found and stored in ADB's. The mode bits of PE's which contain $a_{i,i}$, $a_{i,i+1}$, ... , $a_{i,n}$ are turned on and $c_i \equiv (p_i \pm t_i)a_{ij} + q_i a_{i+1,j} + r_i a_{i+2,j}$ are computed on these PE's. The computation of new elements

$$(i,j) \text{ - element} = a_{ij} - \frac{c_i}{t_i}$$

$$(i,j+1) \text{ - element} = a_{i,j+1} - \frac{c_i q_i}{2K_i^2}$$

$$(i,j+2) \text{ - element} = a_{i,j+2} - \frac{c_i r_i}{2K_i^2}$$

are then performed.

Column Modification:

Similarly $(N_i^* A_k^{(i)})N_i$ is computed from $N_i^* A_k^{(i)}$ for i = m, m + 1, . . . , n in the following way where the element of matrix $N_i^* A_i^{(k)}$ are denoted as $a_{j,h}$:

$$(j,i) \text{ - element} = a_{ji} - [(p_i \pm t_i)a_{ji} + q_i a_{j,i+1} + r_i a_{j,i+2}] \frac{1}{t_i}$$

$$(j,i+1) \text{ - element} = a_{j,i+1} - [(p_i \pm t_i)a_{ji} + q_i a_{j,i+1} + r_i a_{j,i+2}] \frac{q_i}{2K_i^2}$$

$$(j,i+2) \text{ - element} = a_{j,i+2} - [(p_i \pm t_i)a_{ji} + q_i a_{j,i+1} + r_i a_{j,i+2}] \frac{r_i}{2K_i^2}$$

$$\text{for } j = \ell, \ldots, \min [i+3,n].$$

As in the row modification, $c_i' = (p_i \pm t_i)a_{ji} + q_i a_{j,i+1} + r_i a_{j,i+2}$ are computed on the PE's which contain a_{ji} for j from ℓ through min [i + 3, n], then the computations of new (j,i) - element, (j, i + 1) - element and (j, i + 2) - element are performed on the corresponding PE's.

5.5 Computation of eigenvalues

The eigenvalues are calculated as the last step of program after $a_{i+1,i}$ or $a_{i,i-1}$ $a_{i+1,i}$ become negligibly small for all $0<i<n-2$. At each

time when the matrix is deflated by 2, 1 is placed in position n of ADB named SOLV. If the transformation is carried out successfully, SOLV looks like the following,

	0	1	2	3	4	5	6	7	8	9	10	.....	62	63
SOLV	0	0	1	0	0	0	0	1	0	0	1	.....	0	1

Here, the eigenvalues we are looking for are a_{00}, a_{33}, a_{44}, a_{55}, a_{88}, $a_{11,11}$, ... , $a_{61,61}$ and eigenvalues of the following 2 x 2 matrices:

$$\begin{bmatrix} a_{11} & \\ & a_{22} \end{bmatrix}, \quad \begin{bmatrix} a_{66} & \\ & a_{77} \end{bmatrix}, \quad \begin{bmatrix} a_{99} & \\ & a_{10,10} \end{bmatrix} \text{ and } \begin{bmatrix} a_{62,62} & \\ & a_{63,63} \end{bmatrix}.$$

These eigenvalues can be calculated simultaneously on the corresponding PE's.

5.6 The number of iteration required

After each deflation, the number of iterations required to find each eigenvalue (or two eigenvalues) appears in ADB named ITS. This is placed as the nth element of the row vector specified by the user as the fifth parameter in the calling sequence. If this parameter is specified as 0, the number of iterations is not stored anywhere.

REFERENCES

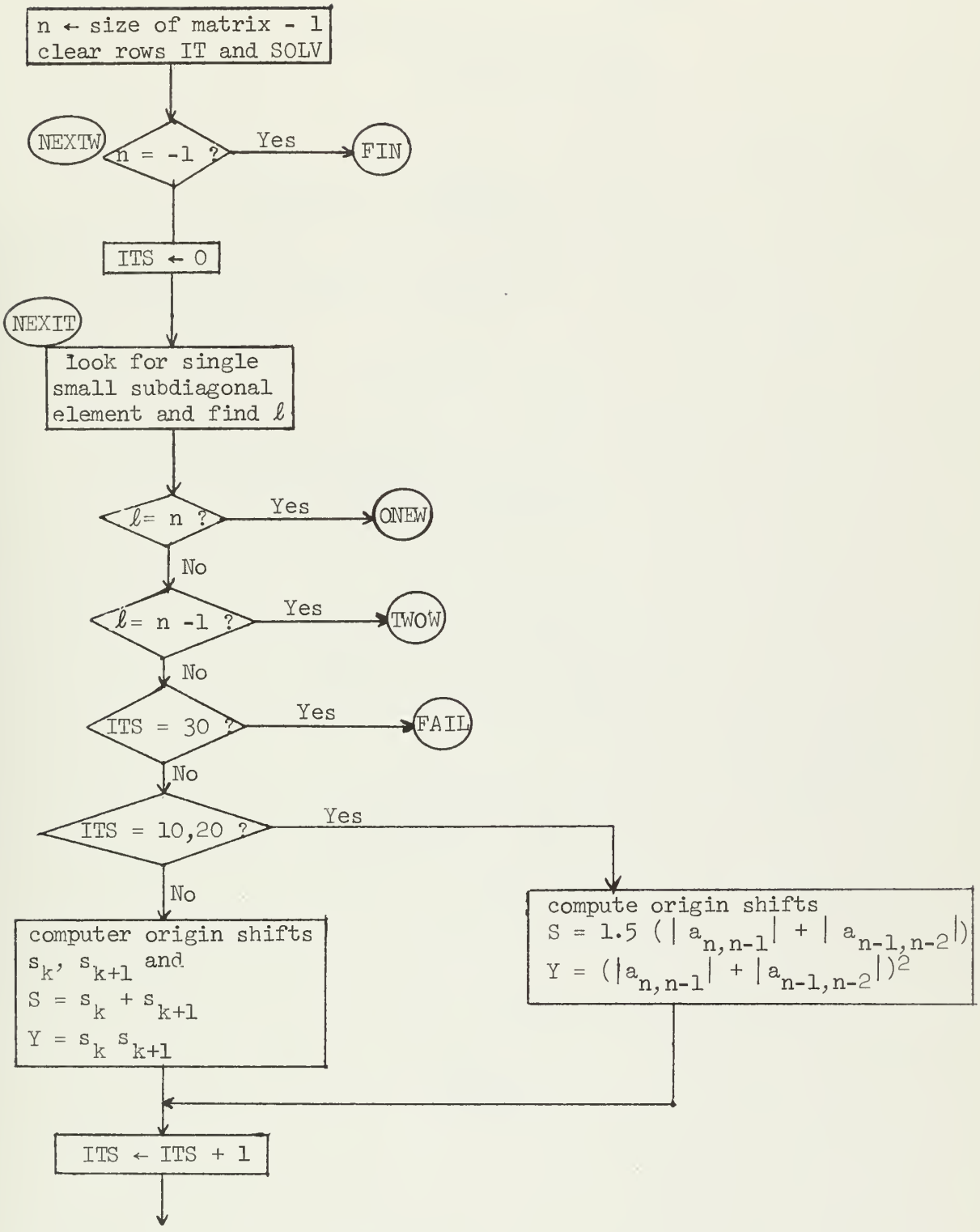
1. J.G.F. Frances: "The QR Transformation--A Unitary Analogue to the LR Transformation," Parts I and II, Comp. Journal, 4, 265-271 and 332-345 (1961/62).

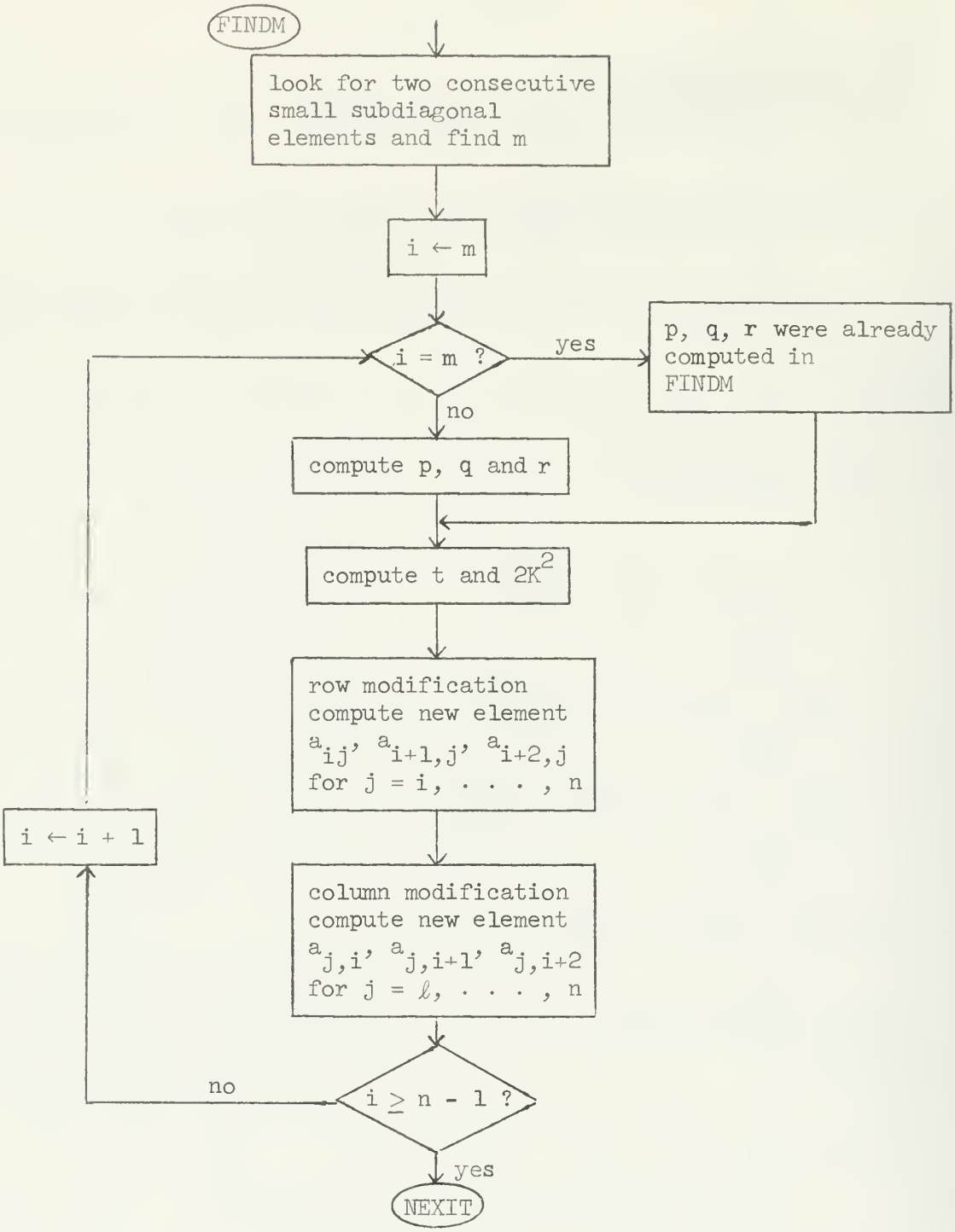
2. R.S. Martin, G. Peters and J.H. Wilkinson: The QR Algorithm for Real Hessenberg Matrices. Numer. Math., 14, 219-231 (1970).

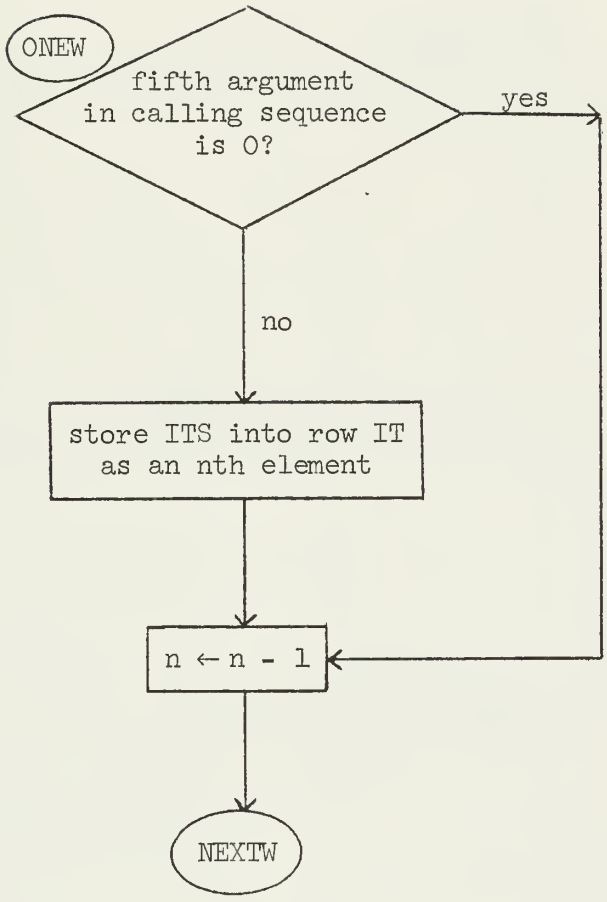
3. B.N. Parlett: The LU and QR Algorithm. Mathematical Methods for Digital Computers, Vol. 2, A. Ralston and H. Wilf, Editors, J. Wiley, 1968.

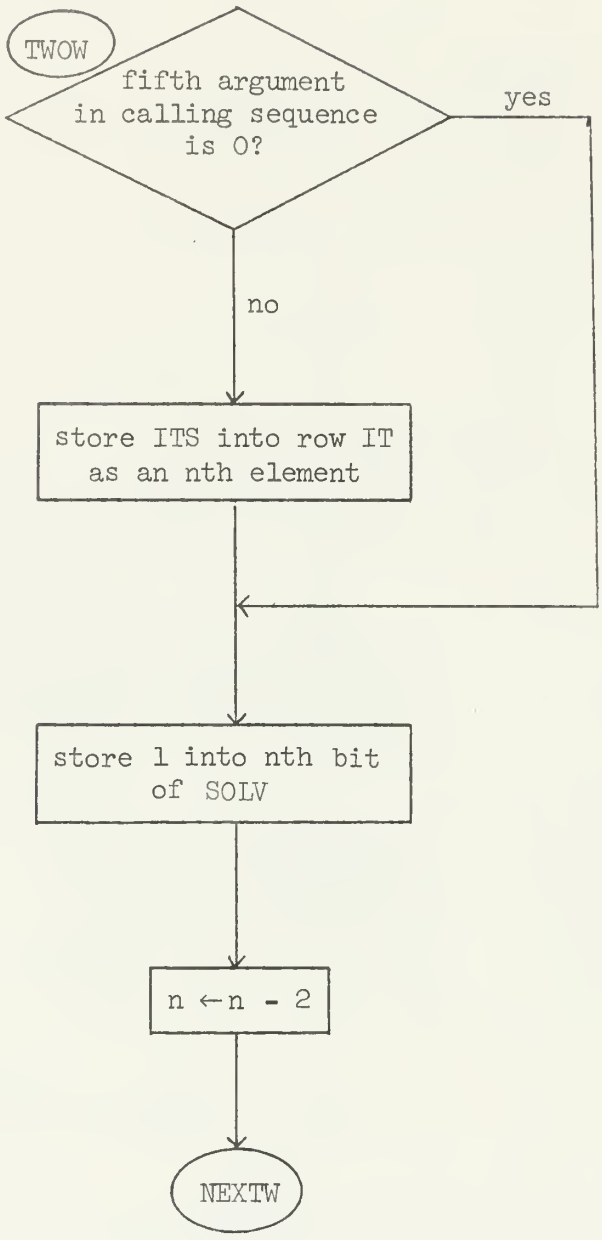
4. J.H. Wilkinson: The Algebraic Eigenvalue Problem. Oxford University Press (1965).

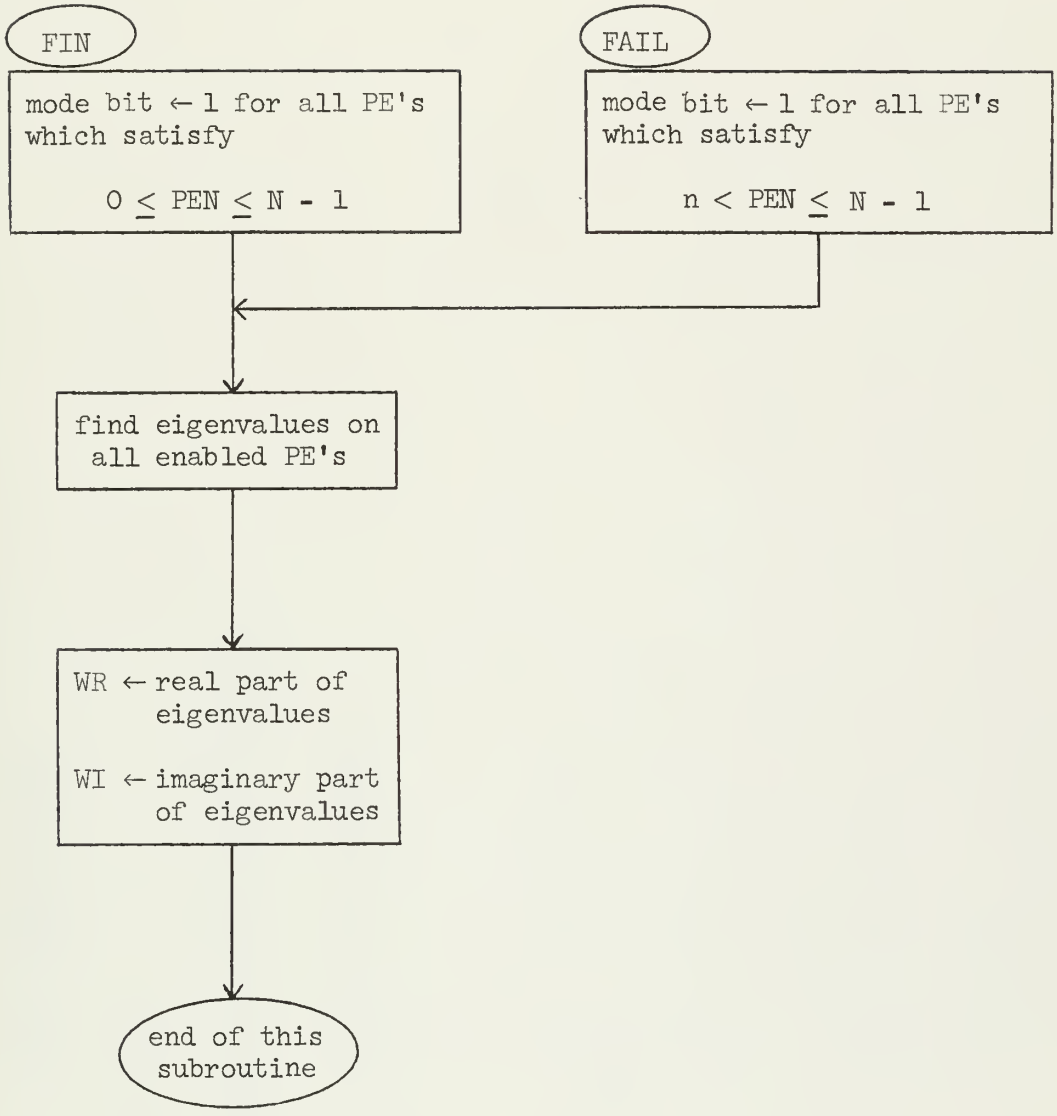
APPENDIX 1


n ← size of matrix - 1
clear rows IT and SOLV
NEXTW
n = -1 ?
Yes
FIN
ITS ← 0
NEXIT
look for single small subdiagonal element and find ℓ
ℓ = n ?
Yes
ONEW
No
ℓ = n -1 ?
Yes
TWOW
No
ITS = 30 ?
Yes
FAIL
No
ITS = 10,20 ?
Yes
No
computer origin shifts
s_k, s_{k+1} and
$S = s_k + s_{k+1}$
$Y = s_k s_{k+1}$
compute origin shifts
$S = 1.5 (|a_{n,n-1}| + |a_{n-1,n-2}|)$
$Y = (|a_{n,n-1}| + |a_{n-1,n-2}|)^2$
ITS ← ITS + 1


FINDM
look for two consecutive small subdiagonal elements and find m
i ← m
i = m ?
yes
p, q, r were already computed in FINDM
no
compute p, q and r
compute t and $2K^2$
row modification
compute new element
a_{ij}, $a_{i+1,j}$, $a_{i+2,j}$
for j = i, . . . , n
column modification
compute new element
$a_{j,i}$, $a_{j,i+1}$, $a_{j,i+2}$
for j = ℓ, . . . , n
i ← i + 1
i ≥ n - 1 ?
no
yes
NEXIT


ONEW
fifth argument in calling sequence is 0?
yes
no
store ITS into row IT as an nth element
n ← n - 1
NEXTW


TWOW
fifth argument in calling sequence is 0?
yes
no
store ITS into row IT as an nth element
store 1 into nth bit of SOLV
n ← n - 2
NEXTW


FIN
mode bit ← 1 for all PE's which satisfy
0 ≤ PEN ≤ N - 1
FAIL
mode bit ← 1 for all PE's which satisfy
n < PEN ≤ N - 1
find eigenvalues on all enabled PE's
WR ← real part of eigenvalues
WI ← imaginary part of eigenvalues
end of this subroutine

```
        BEGIN
.HQSVO:EQU        $D0;
.HQSV1:EQU        $D1;
.HQSV3:EQU        $D2;
.HQSVE:EQU        $D3;
.N    : EQU       $D4;
.H    : EQU       $D5;
.WR   : EQU       $D6;
.WI   : EQU       $D7;
.IT   : EQU       $D8;
.MASK1:EQU        $D9;
.ONE  : EQU       $D10;
.ITS  : EQU       $D11;
.L    : EQU       $D12;
.M    : EQU       $D13;
.SOLV: EQU        $D14;
.NOTLAST:EQU      $D15;
.S    : EQU       $D16;
.Y    : EQU       $D17;
.P    : EQU       $D18;
.Q    : EQU       $D19;
.R    : EQU       $D20;
.XX   : EQU       $D21;
.X    : EQU       $D22;
.Z    : EQU       $D23;
.JBIT: EQU        $D24;
.J    : EQU       $D25;
.SAVI: EQU        $D26;
.SAVJ: EQU        $D27;
.SVO:  EQU        $D28;
.SV1:  EQU        $D29;
%
TMP1 : BLK        1;
TMP2 : BLK        1;
TMP3 : BLK        1;
P    : EQU        TMP1;
Q    : EQU        TMP2;
R    : EQU        TMP3;
       EXTERNAL   SQRT;
       FILL       128;
PEN  : DATA       0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21
                  ,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,
                  40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,
                  58,59,60,61,62,63;
EPS  : DATA       1.0@-10;
%  MATRIX IS ASSUMED TO BE IN STRAIGHT STORAGE
%   STORE PARAMETERS OF SUBROUTINE INTO ADBS
HQR[ENTRY]::
       STL(3)     .HQSV3;              % SAVE CONTENT OF ACAR3
       STL(1)     .HQSV1;              % SAVE CONTENT OF ACAR1
       STL(0)     .HQSV0;              % SAVE CONTENT OF ACAR0
       SETC(3)    E;
       STL(3)     .HQSVE;              % SAVE E BIT
       LOAD(2)    $C0;
       LIT(1)     =1;
```

```
        STL(1)    .ONE;                 % .ONE:=1                                  00000056
        CSUB(0)   $C1;                                                            00000057
        STL(0)    .N;                   %.N:=(SIZE OF MATRIX)-1                   00000058
        ALIT(2)   =1;                                                             00000059
        LOAD(2)   $C0;                                                            00000060
        CSHR(0)   6;                                                              00000061
        STL(0)    .H;                   %H:=PE ADDRESS OF H                       00000062
        ALIT(2)   =1;                                                             00000063
        LOAD(2)   $C0;                                                            00000064
        CSHR(0)   6;                                                              00000065
        STL(0)    .WR;                  %.WR:=PE ADDRESS OF WR                    00000066
        ALIT(2)   =1;                                                             00000067
        LOAD(2)   $C0;                                                            00000068
        CSHR(0)   6;                                                              00000069
        STL(0)    .WI;                  %.WI:=PE ADDRESS OF WI                    00000070
        ALIT(2)   =1;                                                             00000071
        LOAD(2)   $C0;                                                            00000072
        CSHR(0)   6;                                                              00000073
        STL(0)    .IT;                  %.IT:=PE ADDRESS OF IT                    00000074
%       INITIALIZATION                                                            00000075
        LIT(3)    =00000000FFFFFFFF;16;                                           00000076
        STL(3)    .MASK1;                                                         00000077
        CLC(1)    ;                                                               00000078
        STL(1)    .SOLV;                % CLEAR CONTENT OF .SOLV                  00000079
        LDL(0)    .N;                   %ACARO:=(SIZE OF MATRIX)-1                00000080
%  BEGINNING OF N-LOOP; ACARO CONTAINS N                                          00000081
NEXTW:LIT(3)    0.0,-1;                                                           00000082
        CSUB(3)   $C0;                  % ACAR3:=-1-N                            00000083
        SETE      E OR -E;                                                        00000084
        SETE1     E OR E;                                                         00000085
        ZERF(3)   ,NEXW1;                                                         00000086
        JUMP      FIN;                  % IF N=-1 GO TO FIN                      00000087
NEXW1:  LIT(2)    =1,30.0;                                                        00000088
        STL(2)    .ITS;         %ITERATION COUNT .ITS IS SET TO 0                 00000089
% DOUBLE QR ITERATION LOOP                                                       00000090
%  LOOK FOR SINGLE SMALL DIAGONAL ELEMENT                                         00000091
NEXTIT:   LDX     PEN;          % LOAD PE NUMBER TO RGX                          00000092
        IXL       =0(0);                                                          00000093
        JXE       =0(0);                                                          00000094
        SETE      I AND E;                                                        00000095
        SETH      J OR E;               %H:=1 IF RGX LEQ N                        00000096
        SETE      E OR -E;                                                        00000097
        SETE      H AND E;                                                        00000098
        SETE1     E OR E;               %E:=1 IF RGX LEQ N                        00000099
        LDL(3)    .H;                                                             00000100
        LDA       *0(3);                %$A:=H[L,L]                               00000101
        SAP       ;                     % TAKE ABSOLUTE VALUE                    00000102
        RTL       $A,1;                 %$R:=H[L-1,L-1]                           00000103
        ADRN      $R;                                                             00000104
        SLIT(3)   EPS;                                                            00000105
        LOAD(3)   $C3;                  % ACAR3:=EPS                             00000106
        MLRN      $C3;                                                            00000107
        LDS       $A;                   %RGS:=EPS*(ABS(H[L-1,L-1])                00000108
        LDL(3)    .H;                                                             00000109
```

```
        LDA     *1(3);                  % LDA  *H+1                              00000110
        SAP     ;                       %RGA:=ABS(H[L,L-1])                     00000111
        RTL     $A,1;                                                           00000112
        LDA     $R;                                                             00000113
        SBRN    $S;                     %I:=1  IF RGA IS NEGATIVE               00000114
        ISN     ;                                                               00000115
        LIT(1)  =-1,1,0;                                                        00000116
        LDL(3)  .MASK1;                                                         00000117
        CAND(3) $C0;                    % ACAR3:=0,0,N                         00000118
        COR(1)  $C3;                    %ACAR1:=-1,1,N                        00000119
        SETC(2) I;                      % ACAR2:=I                             00000120
BTO :   CTSBT(2) 0(1),CONT1;            % SKIP IF N-TH BIT OF $C2 IS 1          00000121
        TXGTM(1) ,BTO;                                                          00000122
        CLC(1)  ;                       %$C1:=0 IF NO SUBDIAG ELE IS          00000123
%                                         SMALL                                 00000124
CONT1:  LDL(3)  .MASK1;                                                         00000125
        CAND(1) $C3;                    % $C1:=L                               00000126
        CAND(3) $C0;                    % $C3:=N                               00000127
        CSUB(3) $C1;                    %$C3:=N-L                             00000128
        ZERT(3) ,ONEW;                  % IF L=N, GO TO ONEW                    00000129
        CSUB(3) .ONE;                                                           00000130
        ZERT(3) ,TWOW;                  % IF L=N-1, GO TO TWOW                  00000131
        SKIP    ,AFTJ;                                                          00000132
ONEW :  LDL(1)  .IT;            % ONE ROOT IS FOUND                             00000133
        ZERT(1) ,ONEW1;                 % IF IT=0, NO ITERATION COUNT           00000134
        LDL(2)  .ITS;                   %       IS RECORDED                   00000135
        LDL(3)  .MASK1;                                                         00000136
        CAND(2) $C3;                    % $C2:=ITERATION COUNT                  00000137
        SETE    E OR -E;                                                        00000138
        LDX     PEN;                                                            00000139
        IXE     =0(0);                                                          00000140
        SETE    I AND E;                                                        00000141
        SETE1   E OR E;                                                         00000142
        LDA     $C2;                                                            00000143
        STA     0(1);               %IF IT IS NOT 0, IT:=ITERATION COUNT        00000144
ONEW1:  LIT(3)  0,0,-1;                                                         00000145
        CADD(0) $C3;                    % DECREASE ACAR0 BY 1                  00000146
        JUMP    NEXTW;                  %           AND GO TO NEXTW           00000147
TWOW :  LDL(1)  .IT;            % TWO ROOTS ARE FOUND                           00000148
        ZERT(1) ,TWOW1;                 % IF IT=0, NO ITERATION COUNT           00000149
        LDL(2)  .ITS;                   %       IS RECORDED                   00000150
        LDL(3)  .MASK1;                                                         00000151
        CAND(2) $C3;                    % $C2:=ITERATION COUNT                  00000152
        SETE    E OR -E;                                                        00000153
        LDX     PEN;                                                            00000154
        IXE     =0(0);                                                          00000155
        SETE    I AND E;                                                        00000156
        SETE1   E OR E;                                                         00000157
        LDA     $C2;                                                            00000158
        STA     0(1);               %IF IT IS NOT 0, IT:=ITERATION COUNT        00000159
TWOW1:  LDL(2)  .SOLV;                                                          00000160
        CSB(2)  0(0);                                                           00000161
        STL(2)  .SOLV;                  % N-TH BIT OF .SOLV:=1                 00000162
        LIT(3)  0,0,-2;                                                         00000163
        CADD(0) $C3;                    % DECREASE ACAR0 BY 2                  00000164
```

```
        JUMP    NEXTW;                  %       AND GO TO NEXTW                 00000165
AFTJ :  CAND(1) .MASK1;                 % .L:=L                                  00000166
        STL(1)  .L;                                                             00000167
        LDL(2)  .ITS;                                                           00000168
        TXLF(2) $C2,FAILJ;              % IF ITS=30 GO TO FAIL                  00000169
        CAND(2) .MASK1;                                                         00000170
        LIT(1)  =10;                                                            00000171
        CSUB(2) $C1;                                                            00000172
        ZERT(2) ,IT10;                  % IF ITS=10, GO TO IT10                 00000173
        CSUB(2) $C1;                                                            00000174
        ZERT(2) ,IT10;                  % IF ITS=20, GO TO IT10                 00000175
        SKIP    ,ITN10;             % IF ITS NEQ 10,20,30, GO TO ITN10          00000176
FAILJ:  JUMP    FAIL;                                                           00000177
%  FORM SHIFT  S AND Y WHEN IT=10,20                                            00000178
IT10 :  LDL(3)  .H;                                                             00000179
        CADD(3) $C0;                                                            00000180
        LDA     0(3);               % LDA H(0)                                 00000181
        SAP     ;                                                               00000182
        LDS     $A;                                                             00000183
        LDA     -1(3);              % LDA  H=1(0)                              00000184
        SAP     ;                                                               00000185
        RTL     $A,1;                                                           00000186
        LDA     $R;                                                             00000187
        ADRN    $S;                                                             00000188
        LDS     $A;                 %RGS:=ABS(H[N,N-1])                       00000189
%                                               +ABS(H[N-1,N-2])                00000190
        LIT(3)  =1.5;                                                           00000191
        MLRN    $C3;                                                            00000192
        IXE     =-1(0);                                                         00000193
        SETE    I AND E;            %E:=1 IF RGX=N-1                          00000194
        SETE    E OR E;                                                         00000195
        LDC(2)  $A;                                                             00000196
        STL(2)  .S;                 %.S:=1.5*RGS                              00000197
        LDA     $S;                                                             00000198
        MLRN    $S;                                                             00000199
        LDC(3)  $A;                                                             00000200
        STL(3)  .Y;                 %.Y:=RGS**2                               00000201
        SKIP    ,CONT15;                                                        00000202
%  FORM SHIFT S AND Y WHEN IT NEQ 10,20                                         00000203
ITN10:  LDL(3)  .H;                                                             00000204
        CADD(3) $C0;                                                            00000205
        LDA     0(3);               % LDA  H(0)                                00000206
        LDS     -1(3);              % LDS  H-1(0)                              00000207
        RTL     $S,1;                                                           00000208
        IXE     =0(0);                                                          00000209
        SETG    I AND E;            %G:=1 FOR RGX=N                           00000210
        SETE    I AND E;            %E:=G                                     00000211
        SETE1   E OR E;                                                         00000212
        ADRN    $R;                                                             00000213
        LDC(2)  $A;                                                             00000214
        STL(2)  .S;                 %.S:=H[N,N]+H[N-1,N-1]                    00000215
        SETE    E OR -E;                                                        00000216
        SETE    H AND E;            %E:=1 IF RGX LEQ N                        00000217
        SETE1   E OR E;                                                         00000218
```

```
        LDL(3)    .H;                                                    00000219
        CADD(3)   $C0;                                                   00000220
        LDA       0(3);               % LDA  H(0)                       00000221
        MLRN      $R;                                                    00000222
        LDS       $A;                                                    00000223
        LDA       0(3);               % LDA  H(0)                       00000224
        RTL       $A,1;                                                  00000225
        LDA       $R;                                                    00000226
        SETE      G AND E;                                               00000227
        SETE1     E OR E;                                                00000228
        MLRN      -1(3);              % MLRN H-1(0)                     00000229
        CHSA      ;                                                      00000230
        ADRN      $S;                                                    00000231
        LDC(3)    $A;                                                    00000232
        STL(3)    .Y;                 %,Y;=H[N,N]*H[N-1,N-1]           00000233
%                                                 -H[N,N-1]*H[N-1,N]     00000234
%  LOOK FOR TWO CONSECUTIVE SMALL DIAGONAL ELEMENT FOR M=N-2 TO L        00000235
CONT15;LDL(1)     .ITS;                                                  00000236
        LDL(2)    .L;                                                    00000237
        ALIT(1)   =1;                 % INCREASE ITERATION COUNT BY 1   00000238
        STL(1)    .ITS;               % AND STORE IN .ITS               00000239
        JXG       =0(2);                                                 00000240
        IXL       =-1(0);                                                00000241
        SETE      E OR -E;                                               00000242
        SETE      J AND E;                                               00000243
        JXE       =0(2);                                                 00000244
        SETE      J OR E;                                                00000245
        SETG      I AND E;            %G;=1 IF L LEQ RGX LEQ N-2         00000246
%  COMPUTE P=H[M,M]*(H[M,M]-S)+Y+H[M+1,M]*H[M,M+1]                       00000247
        SETE      E OR -E;                                               00000248
        SETE      H AND E;            %E;=1 IF RGX LEQ N                 00000249
        SETE1     E OR E;                                                00000250
        LDL(3)    .H;                                                    00000251
        LDA       *0(3);              % LDA  *H                         00000252
        LDS       $A;                                                    00000253
        LDL(3)    .S;                                                    00000254
        SBRN      $C3;                                                   00000255
        SETE      G AND E;                                               00000256
        SETE1     E OR E;                                                00000257
        MLRN      $S;                                                    00000258
        LDS       $A;                                                    00000259
        SETE      E OR -E;                                               00000260
        SETE      H AND E;                                               00000261
        SETE1     E OR E;                                                00000262
        LDL(3)    .H;                                                    00000263
        LDA       *-1(3);             % LDA  *H-1                       00000264
        RTL       $A,-1;                                                 00000265
        LDA       $R;                                                    00000266
        SETE      G AND E;            %E;=1 IF L LEQ RGX LEQ N-2         00000267
        SETE1     E OR E;                                                00000268
        MLRN      *1(3);              % MLRN *H+1                       00000269
        ADRN      $S;                                                    00000270
        LDL(2)    .Y;                                                    00000271
        ADRN      $C2;                                                   00000272
        STA       P;                                                     00000273
```

```
%  COMPUTE Q=H[M+1,M]*(H[M,M]+H[M+1,M+1]-S)                                    00000274
        SETE    E OR -E;                                                     00000275
        SETE    H AND E;            %E:=1 FOR RGX LEQ N                      00000276
        SETE1   E OR E;                                                      00000277
        LDL(3)  ,H;                                                          00000278
        LDA     *0(3);              % LDA  *H                               00000279
        RTL     $A,-1;                                                       00000280
        SETE    G AND E;            %E:=1 FOR L LEQ RGX LEQ N-2              00000281
        SETE1   E OR E;                                                      00000282
        ADRN    $R;                                                          00000283
        LDL(2)  ,S;                                                          00000284
        SBRN    $C2;                                                         00000285
        MLRN    *1(3);              % MLRN  *H+1                            00000286
        STA     Q;                                                           00000287
%  COMPUTE R=H[M+1,M]*H[M+2,M+1]                                             00000288
        SETE    E OR -E;                                                     00000289
        SETE    H AND E;            %E:=1 FOR RGX LEQ N                      00000290
        SETE1   E AND E;                                                     00000291
        LDL(3)  ,H;                                                          00000292
        LDA     *1(3);              % LDA  *H+1                             00000293
        RTL     $A,-1;                                                       00000294
        SETE    G AND E;            %E:=1 FOR L LEQ RGX LEQ N-2              00000295
        SETE1   E OR E;                                                      00000296
        MLRN    $R;                                                          00000297
        STA     R;                                                           00000298
%  COMPUTE  P,Q,R=P,Q,R/(ABS(P)+ABS(Q)+ABS(R))                               00000299
        LDA     P;                                                           00000300
        SAP     ;                   %RGA:=ABS(P)                             00000301
        LDS     $A;                                                          00000302
        LDA     Q;                                                           00000303
        SAP     ;                   %RGA:=ABS(Q)                             00000304
        ADRN    $S;                                                          00000305
        LDS     $A;                                                          00000306
        LDA     R;                                                           00000307
        SAP     ;                                                            00000308
        ADRN    $S;                                                          00000309
        LDS     $A;                 %RGS:=ABS(P)+ABS(Q)+ABS(R)               00000310
        LIT(2)  =1.0;                                                        00000311
        CLRA    ;                                                            00000312
        LDB     $A;                                                          00000313
        LDA     $C2;                %RGA:=1.0, RGB:=0                        00000314
        DVRN    $S;                                                          00000315
        LDS     $A;                 %RGS:=1.0/(ABS(P)+ABS(Q)+ABS(R))         00000316
        LDA     P;                                                           00000317
        MLRN    $S;                                                          00000318
        STA     P;                  %P:=P*RGS                                00000319
        LDA     Q;                                                           00000320
        MLRN    $S;                                                          00000321
        STA     Q;                  %Q:=Q*RGS                                00000322
        LDA     R;                                                           00000323
        MLRN    $S;                                                          00000324
        STA     R;                  %R:=R*RGS                                00000325
%  TEST IF  ABS(H[M,M-1])*(ABS(Q)+ABS(R))                                    00000326
%                 <EPS*(ABS(P)*ABS(H[M-1,M-1])+ABS(X)+ABS(Z))                00000327
```

```
SETE      E OR -E;                                                    00000328
SETE      H AND E;                                                    00000329
SETE1     E OR E;           %E:=1 FOR RGX LEQ N                       00000330
LDA       Q;                                                          00000331
SAP       ;                                                           00000332
LDS       $A;                                                         00000333
LDA       R;                                                          00000334
SAP       ;                                                           00000335
ADRN      $S;                                                         00000336
LDS       $A;               %RGS:=ABS(Q)+ABS(R)                       00000337
LDL(3)    .H;                                                         00000338
LDA       *1(3);            % LDA   *H+1                             00000339
SAP       ;                                                           00000340
RTL       $A,1;                                                       00000341
LDL(2)    .L;                                                         00000342
JXE       =0(2);            %J:=1 FOR RGX=L                         00000343
 SETE     G AND E;                                                    00000344
SETE      -J AND E;         %E:=1 FOR L<RGX LEQ N-2                   00000345
SETI      E OR E;           %I:=1 FOR L<RGX LEQ N-2                   00000346
 SETE1    E OR E;                                                     00000347
LDA       $R;                                                         00000348
MLRN      $<;                                                         00000349
LDL(3)    .WR;                                                        00000350
STA       0(3);             %WR:=(ABS(Q)+ABS(R))                      00000351
SETE      E OR -E;                                                    00000352
SETE      H AND E;          %E:=1 FOR RGX LEQ N                       00000353
SETE1     E OR E;                                                     00000354
LDL(3)    .H;                                                         00000355
LDA       *0(3);            % LDA   *H                               00000356
SAP       ;                                                           00000357
LDS       $A;                                                         00000358
RTL       $S,1;                                                       00000359
SETE      I AND E;          %E:=1 FOR L<RGX LEQ N-2                   00000360
SETE1     E OR E;                                                     00000361
ADRN      $R;                                                         00000362
RTL       $S,-1;                                                      00000363
ADRN      $R;                                                         00000364
LDS       $A;               %RGS:=ABS(H[M+1,M+1])                     00000365
LDA       P;                %     +ABS(H[M,M])+ABS(H[M-1,M-1])        00000366
SAP       ;                                                           00000367
SLIT(2)   EPS;                                                        00000368
LOAD(2)   $C2;                                                        00000369
MLRN      $C2;                                                        00000370
MLRN      $S;               %RGA:=EPS*ABS(P)*RGS                      00000371
CHSA      ;                                                           00000372
LDL(3)    .WR;                                                        00000373
ADRN      0(3);             % ADRN  WR;                              00000374
JSN       ;                 %J:=1 IF RGA IS NEGATIVE                  00000375
LIT(1)    0,0,-2;                                                     00000376
CADD(1)   $C0;                                                        00000377
CAND(1)   .MASK1;           %ACAR1:=0,0,N-2                           00000378
LIT(3)    =-1,0,0;                                                    00000379
COR(1)    $C3;              % ACAR1:=-1,0,N-2                         00000380
LDL(3)    .L;                                                         00000381
CADD(3)   .ONE;                                                       00000382
```

```
        CSHL(3)   24;                    % ACAR3;=0,L+1,0                          00000383
        COR(1)    $C3;                   % ACAR1;=-1,L+1,N-2                        00000384
        SETC(3)   J;                     %ACAR3;=1 IF J CONTAINS 1                00000385
BT1   ; CTSBT(3)  0(1),CONT2;                                                       00000386
        TXGTM(1)  ,BT1;                                                           00000387
        LDL(1)    ,L;                                                             00000388
CONT2;  CAND(1)   ,MASK1;                % ACAR1;=0,0,M                            00000389
        STL(1)    ,M;                    %,M;=0,0,M                               00000390
%  PUT ZERO IN APPROPRIATE PLACES                                                 00000391
        SETE      E OR -E;                                                        00000392
        SETE1     E OR E;                                                         00000393
        CLRA      ;                                                               00000394
        IXG       =0(1);                                                          00000395
        JXE       =0(1);                                                          00000396
        SETE      I AND E;                                                        00000397
        SETE      J OR E;                                                         00000398
        JXL       =-2(0);                                                         00000399
        SETE      J AND E;               % E,E1;=1 FOR M LEQ RGX < N-2             00000400
        SETE1     E OR E;                                                         00000401
        LDL(3)    ,M;                                                             00000402
        STA       +3(3);                 % STA  *M+3                             00000403
        IXE       =-2(0);                                                         00000404
        SETE      I OR E;                % E,E1;=1 FOR M LEQ RGX LEQ N-2           00000405
        SETE1     E OR E;                                                         00000406
        STA       +2(3);                 % STA  *M+2                             00000407
%  PUT P,Q,R INTO ADB WHEN K=M                                                    00000408
        SETE      E OR -E;               %PICK UP M-TH ELEMENT OF                 00000409
        SETE1     E OR E;                % P,Q AND R WHEN K=M                     00000410
        CLRA      ;                                                               00000411
        IXE       =0(1);                 %I;=1 IF RGX=M                           00000412
        SETE      I AND E;               % E,E1;= 1                               00000413
        SETE1     E OR E;                                                         00000414
        LDA       P;                                                              00000415
        LDS       Q;                                                              00000416
        LDC(2)    $A;                                                             00000417
        LDC(3)    $S;                                                             00000418
        STL(2)    ,P;                    %,P;=M-TH ELEMENT OF P                   00000419
        STL(3)    ,Q;                    %,Q;=M-TH ELEMENT OF Q                   00000420
        LDA       R;                                                              00000421
        LDC(2)    $A;                                                             00000422
        STL(2)    ,R;                                                             00000423
%  ARRANGE MATRIX INTO SKEW STORAGE FORM                                          00000424
        LIT(2)    -1,1,0;                                                         00000425
        COR(2)    $C0;                   %ACAR2;=-1,1,N                           00000426
        SETE      E OR -E;                                                        00000427
        SETE1     E OR E;                                                         00000428
DTAR1;  LDL(3)    ,H;                                                             00000429
        CAOD(3)   $C2;                                                            00000430
        LDA       0(3);                  % LDA  H(2)                             00000431
        RTL       $A,0(2);                                                        00000432
        STR       0(3);                  % STR H(2)                              00000433
        TXGTM(2)  ,DTAR1;                                                         00000434
%  DOUBLE QR STEP INVOLVING ROWS L TO N AND COLUMNS M TO N                        00000435
        LIT(3)    =0,1,-1;               %INITIALIZATION OF K-LOOP                00000436
```

```
        CADD(3)   $C0;                    %ACAR3:=0,1,N-1                                  00000437
        CSHL(3)   24;                     %ACAR3:=1,N-1,0                                  00000438
        COR(1)    $C3;                    %ACAR1:=1,N-1,M                                  00000439
%  ACAR1 CONTROLS K-LOOP, KIM STEP 1 UNTIL N-1                                             00000440
NEXTK:  LOL(2)    $C0;                    % SET NOTLAST WITH APPROPRIATE                   00000441
        CSUB(2)   ,ONE;                   % BOOLEAN EXPRESSION                             00000442
        CSUB(2)   $C1;                    %ACAR2:=(N-1)-K                                  00000443
        ZERT(2)   ,LAST;                                                                   00000444
        LOL(3)    ,ONE;                                                                    00000445
        STL(3)    ,NOTLAST;               % NOTLAST:=1 IF K NEQ N-1                        00000446
        SKIP      ,CONT17;                                                                 00000447
LAST:   CLC(3)    ;                                                                        00000448
        STL(3)    ,NOTLAST;               %NOTLAST:=0 IF K=N-1                             00000449
CONT17:                                                                                    00000450
        SETE      E OR -E;                                                                 00000451
        SETE1     E OR E;                                                                  00000452
        CLRA      ;                                                                        00000453
        LOL(3)    ,MASK1;                 % TEST IF K=M                                   00000454
        CAND(3)   $C1;                    % ACAR3:=0,0,K                                  00000455
        CSUB(3)   ,M;                     % ,ACAR3:=K-M                                   00000456
        ZERT(3)   ,KEQM;                  % IF K=M GO TO KEQM,                             00000457
                                          % P,Q,R ARE ALREADY COMPUTED                     00000458
%  COMPUTE P=H[K,K-1],Q=H[K+1,K-1] AND R=H[K+2,K-1] WHEN K NEQ N                           00000459
        LDX       PEN;                                                                     00000460
        RTL       $X,-1(1);                                                                00000461
        LDX       $R;                                                                      00000462
        IXE       =2(1);                                                                   00000463
        SETH      I AND E;                %H:=1 FOR H[K+2,K-1]                             00000464
        JXE       =1(1);                  %J:=1 FOR H[K+1,K-1]                             00000465
        IXE       =0(1);                  %I:=1 FOR H[K,K-1]                               00000466
        SETE      I AND E;                                                                 00000467
        SETE      J OR E;                 %G:=1 FOR H[K,K-1],H[K+1,K-1]                    00000468
        SETG      H OR E;                 %              AND H[K+2,K-1]                    00000469
        SETE      E OR -E;                                                                 00000470
        SETE      G AND E;                                                                 00000471
        SETE1     E OR E;                 % E,E1:=G                                       00000472
        LOL(3)    ,M;                                                                      00000473
        LDA       *0(3);                  % LDA  *H                                       00000474
        LDL(2)    ,NOTLAST;                                                                00000475
        ZERF(2)   ,CPQR;                                                                   00000476
        SETE      H AND E;                                                                 00000477
        SETE1     E OR E;                 % IF NOTLAST=0, H[K+2,K-1]                      00000478
        CLRA      ;                       % IS CONSIDERED TO BE 0                         00000479
        SETE      E OR -E;                                                                 00000480
        SETE      G AND E;                % RESET E,E1                                    00000481
        SETE1     E OR E;                                                                  00000482
CPQR:   LOS       $A;                     %RGS:=H[K,K-1],H[K+1,K-1]                      00000483
                                          %              AND H[K+2,K-1]                    00000484
        SAP       ;                       % RGA:=ABS(RGA)                                 00000485
        RTL       $A,1;                                                                    00000486
        ADRN      $R;                                                                      00000487
        RTL       $A,2;                   %RGA:=ABS(H[K,K-1])+ABS(H[K+1,K-1]             00000488
        ADRN      $R;                     %   +ABS(H[K+2,K-1])) AT H=1                     00000489
        SETE      H AND E;                %E:=H                                          00000490
        SETE1     E OR E;                                                                  00000491
```

```
        LDC(2)    $A;                    %ACAR2:=RGA AT M=1                    00000492
        STL(2)    .XX;                   % .XX:=ACAR2                         00000493
        ZERF(2)   .NXPQ;                                                       00000494
        JUMP      CONT3;                 % IF .XX=0 GO TO CONT3                00000495
NXPQ:   SETE      E OR ~E;                                                     00000496
        SETE      G AND E;               % E,E1:=G                            00000497
        SETE1     E OR E;                                                      00000498
        CLRA      ;                                                            00000499
        LDB       $A;                                                          00000500
        LDA       $S;                                                          00000501
        DVRN      $C2;                                                         00000502
        LDS       $A;                    %RGS:=P/XX,Q/XX,R/XX                 00000503
        SKIP      .CONT27;                                                     00000504
%  PLACE P,Q,R IN APPROPRIATE REGISTER WHEN K=M                                00000505
KEQM:   LDX       PEN;                                                         00000506
        LDL(2)    .MASK1;                                                      00000507
        CAND(2)   $C1;                                                         00000508
        ZERT(2)   .KZERO;                                                      00000509
        RTL       $X,-1(1);                                                    00000510
        SKIP      .KEQM2;                                                      00000511
KZERO:  RTL       $X,0(1);               % DUMMY BIT SETTING FOR COMPUTING     00000512
KEQM2:  LDX       $R;                    %                  S IN CONT27       00000513
        IXE       =2(1);                                                       00000514
        SETH      I AND E;               %H:=1 FOR H[K+2,K]                   00000515
        JXE       =1(1);                 %J:=1 FOR H[K+1,K]                   00000516
        IXE       =0(1);                 %I:=1 FOR H[K,K]                     00000517
        SETE      I AND E;                                                     00000518
        SETE      J OR E;                                                      00000519
        SETG      H OR E;          % G:=1 FOR H[K,K],H[K+1,K],H[K+2,K]        00000520
        SETE      E OR ~E;                                                     00000521
        LDL(2)    .P;                                                          00000522
        LDL(3)    .Q;                                                          00000523
        SETE      I AND E;                                                     00000524
        SETE1     E OR E;                % E,E1:=I                            00000525
        ADRN      $C2;                   %RGA:=0,...,0,P,0,...,0              00000526
        SETE      E OR ~E;                                                     00000527
        SETE      J AND E;                                                     00000528
        SETE1     E OR E;                %E,E1:=J                            00000529
        ADRN      $C3;                   %RGA:=0,...,0,P,Q,0,...,0            00000530
        LDL(2)    .R;                                                          00000531
        SETE      E OR ~E;                                                     00000532
        SETE      H AND E;                                                     00000533
        SETE1     E OR E;                %E,E1:=H                            00000534
        ADRN      $C2;                   %RGS,RGA:=0,...,0,P,Q,R,0...,0       00000535
        SETE      E OR ~E;                                                     00000536
        SETE1     E OR E;                                                      00000537
        LDS       $A;                                                          00000538
%  COMPUTE S=SQRT(P**2+Q**2+R**2)                                              00000539
CONT27;                                                                        00000540
        SETC(2)   J;                                                           00000541
        STL(2)    .JBIT;                 %SAVE J BIT INTO .JBIT               00000542
        JSN       ;                      %J:=SIGN BIT OF RGA                  00000543
        SETE      I AND E;                                                     00000544
        SETJ      J AND E;               %J:=1 IF P IS NEGATIVE               00000545
```

```
      SETE      E OR -E;                                                    00000546
      SETE      G AND E;              % E,E1:=G                             00000547
      SETE1     E OR E;                                                     00000548
      MLRN      $A;                 %$A:=(P/XX)**2,(Q/XX)**2,(R/XX)**2      00000549
      RTL       $A,1;                                                       00000550
      ADRN      $R;                                                         00000551
      RTL       $A,2;                                                       00000552
      ADRN      $R;                 %$A:=(P/XX)**2+(Q/XX)**2+(R/XX)**2      00000553
      SETE      E OR -E;                                                    00000554
      SETE      H AND E;                                                    00000555
      SETE1     E OR E;                                                     00000556
      STL(0)    .SV0;                 % SAVE ACAR0 INTO HQSV0              00000557
      STL(1)    .SV1;                 % SAVE ACAR1 INTO HQSV1              00000558
      SETC(0)   I;                    % SAVE I-BIT                         00000559
      SETC(1)   J;                    % SAVE J-BIT                         00000560
      STL(0)    .SAVI;                                                      00000561
      STL(1)    .SAVJ;                                                      00000562
      CLC(3)    ;                                                           00000563
      SLIT(3)   SQRT;                 % CALL SQRT SUBROUTINE               00000564
      EXCHL(3)  $ICR;                 %$A:=SQRT($A)                       00000565
      LDL(0)    .SAVI;                                                      00000566
      LDL(1)    .SAVJ;                                                      00000567
      LDI       $C0;                                                        00000568
      LDJ       $C1;                                                        00000569
      LDL(0)    .SV0;                                                       00000570
      LDL(1)    .SV1;                 % RESTORE ACAR0 AND ACAR0            00000571
      SETC(2)   J;                    % ACAR2:=J BIT                       00000572
      ZERT(2)   ,PPOS;                %SKIP TO PPOS IF ACAR2 HAS ALL 0    00000573
      CHSA      ;                    %RGA:=-RGA IF J BIT HAS 1            00000574
PPOS : LDC(2)   $A;                                                       00000575
      STL(2)    .S;                   %.S:=SQRT((P/XX)**2+(Q/XX)**2       00000576
                                      %                  +(R/XX)**2)      00000577
      LDL(3)    .JBIT;                                                      00000578
      LDJ       $C3;                  % RESTORE J-BIT WITH .JBIT          00000579
%  COMPUTE X,Y,Z FOR ROW AND COLUMN MODIFICATION                           00000580
      SETE      E OR -E;                                                    00000581
      SETE      I AND E;              % E,E1:=I                            00000582
      SETE1     E OR E;                                                     00000583
      LDA       $S;                                                         00000584
      ADRN      $C2;                  %$A:=P+.S WHERE I=1                 00000585
      LDS       $A;                                                         00000586
      LDC(3)    $A;                                                         00000587
      STL(3)    .P;                   %.P:=.P+.S                          00000588
      SETE      E OR -E;                                                    00000589
      SETE      G AND E;                                                    00000590
      SETE1     E OR E;               % E,E1:=G                            00000591
      CLRA      ;                                                           00000592
      LDB       $A;                                                         00000593
      LDA       $S;                                                         00000594
      DVRN      $C2;                  %RGA:=.P/.S,.Q/.S,.R/.S             00000595
      SETE      I AND E;              % E,E1:=I                            00000596
      SETE1     E OR E;                                                     00000597
      LDC(3)    $A;                                                         00000598
      STL(3)    .X;                   %.X:=.P/.S                          00000599
      SETE      E OR -E;                                                    00000600
```

```
        SETE    J AND E;                % E,E1:=J                                  00000601
        SETE1   E OR E;                                                           00000602
        LDC(2)  $A;                                                               00000603
        STL(2)  .Y;                     %,Y:=,Q/,S                                00000604
        SETE    E OR -E;                                                          00000605
        SETE    H AND E;                % E,E1:=H                                  00000606
        SETE1   E OR E;                                                           00000607
        LDC(3)  $A;                                                               00000608
        STL(3)  .Z;                     %,Z:=,R/,S                                00000609
%  COMPUTE Q=Q/P AND R=R/P FOR ROW AND COLUMN MODIFICATION                        00000610
        SETE    E OR -E;                                                          00000611
        SETE    J AND E;                                                          00000612
        SETE    H OR E;                 % E,E1:=J OR H                            00000613
        SETE1   E OR E;                                                           00000614
        LDL(3)  .P;                                                               00000615
        CLRA    ;                                                                 00000616
        LDB     $A;                                                               00000617
        LDA     $S;                                                               00000618
        DVRN    $C3;                    %RGA:=,Q/,P,,R/,P FOR J=1 AND H=1         00000619
        SETE    J AND E;                                                          00000620
        SETE1   E OR E;                                                           00000621
        LDC(2)  $A;                                                               00000622
        STL(2)  .Q;                     %,Q:=,Q/,P                                00000623
        SETE    E OR -E;                                                          00000624
        SETE    H AND E;                                                          00000625
        SETE1   E OR E;                                                           00000626
        LDC(3)  $A;                                                               00000627
        STL(3)  .R;                     %,R:=,R/,P                                00000628
%  COMPUTE NEW H[K,K-1]                                                           00000629
%   IF K NEQ M THEN H[K,K-1]:= -S*X                                               00000630
%           ELSE IF L NEQ M THEN H[K,K-1]:= -H[K,K-1]                             00000631
        LDL(2)  .MASK1;                                                           00000632
        CAND(2) $C1;                    %ACAR2:=0,0,K                             00000633
        CSUB(2) .M;                     %ACAR2:=0,0,K-M                           00000634
        ZERT(2) .CPLM;                  % IF K=M SKIP TO COMPARISON               00000635
        SETE    E OR -E;                %   BETWEEN L AND M                       00000636
        SETE    I AND E;                                                          00000637
        SETE1   E OR E;                 % E,E1:=I                                 00000638
        CLRA    ;                     %IF K NEQ M, H[K,K-1]:=-S*X                 00000639
        LDL(2)  .S;                                                               00000640
        LDL(3)  .XX;                                                              00000641
        SBRN    $C2;                                                              00000642
        MLRN    $C3;                                                              00000643
        LDL(3)  .H;                                                               00000644
        STA     *0(3);                  % STA  *H                                 00000645
        SKIP    .RWMOD;                                                           00000646
CPLM :  LDL(2)  .L;                                                               00000647
        CSUB(2) .M;                     % TEST IF L=M                             00000648
        ZERT(2) .RWMOD;               %H[K,K-1]:=-H[K,K-1] IF L NEQ M,K=M         00000649
        LDX     PEN;                                                              00000650
        RTL     $X,-1(1);               % ACAR1 CONTAINS M WHICH IS               00000651
        LDX     $R;                     %         GREATER THAN L                  00000652
        IXE     =0(1);                                                            00000653
        SETE    E OR -E;                                                          00000654
```

```
      SETE      I AND E;                                                    00000655
      SETE1     E OR E;                                                     00000656
      LDL(3)    .H;                                                         00000657
      LDA       *0(3);              % LDA  *H                              00000658
      CHSA      ;                                                           00000659
      STA       *0(3);              % STA  *H                              00000660
%                                                                         00000661
   DEFINE  KRGXN =                                                        00000662
      IXG       =0(1);              %I:=1 IF K<RGX                        00000663
      JXL       =0(0);              %J:=1 IF RGX<N                        00000664
      SETE      I AND E;                                                  00000665
      SETE      J AND E;            %E:=1 IF K<RGX<N                      00000666
      IXE       =0(1);              %I:=1 IF RGX=K                        00000667
      JXE       =0(0);              %J:=1 IF RGX=N                        00000668
      SETE      I OR E;         % E,E1 ARE SET WHEN K LEQ RGX LEQ N         00000669
      SETE      J OR E ##;                                                00000670
%                                                                         00000671
RWMOD: SETE     E OR -E;                                                  00000672
%  ROW MODIFICATION                                                       00000673
      LDX       PEN;                                                      00000674
      RTL       $X,2(1);                                                  00000675
      LDX       $R;                                                       00000676
      KRGXN     ;                                                         00000677
      SETH      E OR E;             % H IS SET FOR H[K+2,J]                00000678
      SETE      E OR -E;                                                  00000679
      RTL       $X,-1;                                                    00000680
      LDX       $R;                                                       00000681
      KRGXN     ;                                                         00000682
      SETG      E OR E ;            % G IS SET FOR H[K+1,J]                00000683
      RTL       $X,-1;                                                    00000684
      SETE      E OR -E;                                                  00000685
      LDX       $R;                                                       00000686
      KRGXN     ;                                                         00000687
      SETJ      E OR E;             % J IS SET FOR H[K,J]                  00000688
      SETE      E OR -E;                                                  00000689
      SETE1     E OR E;                                                   00000690
      LDL(3)    .H;                                                       00000691
      CADD(3)   $C1;                                                      00000692
      LDA       1(3);               % LDA H+1(1),  RGA:=H+1(1)            00000693
      SETE      G AND E;            % E,E1:=G                              00000694
      SETE1     E OR E;                                                   00000695
      LDL(2)    .Q;                                                       00000696
      MLRN      $C2;                                                      00000697
      LDS       $A;                 % RGS:=.Q*RGA                         00000698
      SETE      E OR -E;                                                  00000699
      SETE1     E OR E;             % E,E1:=1                              00000700
      LDA       0(3);               % LDA  H(1)                           00000701
      RTL       $A,1;               % ROUTE RGA BY 1                      00000702
      LDA       $R;                                                       00000703
      ADRN      $S;                                                       00000704
      LDS       $A;                 % RGS:=RGS+RGR                        00000705
      LDL(2)    .NOTLAST;                                                 00000706
      ZERT(2)   .CONT4;             % IF NOTLAST=0 GO TO CONT4            00000707
%  IF NOTLAST=1  COMPUTE                                                  00000708
%    H[K+2,J]:= H[K+2,J]-(H[K,J]+Q*H[K+1,J]+R*H[K+2,J])*Z                 00000709
```

```
        LDA     2(3);                   %RGA:=H+2(1)                    00000710
        RTL     $S,1;                   % ROUTE RGS BY 1                00000711
        LDS     $R;                                                     00000712
        SETE    H AND E;                                                00000713
        SETE1   E OR E;                 % E,E1:=H                       00000714
        LDL(2)  .R;                                                     00000715
        LDL(3)  .Z;                                                     00000716
        MLRN    $C2;                                                    00000717
        ADRN    $S;                                                     00000718
        LDS     $A;                     % RGS:=RGA*.R+RGS               00000719
        MLRN    $C3;                    % RGA:=.Z*RGS                   00000720
        CHSA    ;                                                       00000721
        LDL(3)  .H;                                                     00000722
        CADD(3) $C1;                                                    00000723
        ADRN    2(3);                   % ADRN  H+2(1)                  00000724
        STA     2(3);                   %H+2(1):=H+2(1)-RGA            00000725
        SETE    E OR -E;                                                00000726
        SETE1   E OR E;                                                 00000727
        RTL     $S,-1;                                                  00000728
        LDS     $R;                     % ROUTE RGS BY -1               00000729
%  COMPUTE                                                              00000730
%   H[K+1,J]:= H[K+1,J]-(H[K,J]+Q*H[K+1,J]+R*H[K+2,J])*Y                00000731
CONT4:  SETE    E OR -E;                                                00000732
        SETE    G AND E;                                                00000733
        SETE1   E OR E;                 % E,E1:=G                       00000734
        LDL(2)  .Y;                                                     00000735
        LDA     $S;                                                     00000736
        CHSA    ;                                                       00000737
        MLRN    $C2;                    % RGA:=-RGS*.Y                  00000738
        LDL(3)  ,H;                                                     00000739
        CADD(3) $C1;                                                    00000740
        ADRN    1(3);                   % ADRN  H+1(1)                  00000741
        STA     1(3);                   % STA   H+1(1)                  00000742
%  COMPUTE                                                              00000743
%    H[K,J]:= H[K,J]-(H[K,J]+Q*H[K+1,J]+R*H[K+2,J])*X                   00000744
        SETE    E OR -E;                                                00000745
        SETE    J AND E;                                                00000746
        SETE1   E OR E;                 % E,E1:=J                       00000747
        RTL     $S,-1;                  % ROUTE RGS BY -1               00000748
        LDL(3)  ,X;                                                     00000749
        LDA     $R;                                                     00000750
        CHSA    ;                                                       00000751
        MLRN    $C3;                    %RGA:=-RGS*,X                  00000752
        LDL(3)  ,H;                                                     00000753
        CADD(3) $C1;                                                    00000754
        ADRN    0(3);                   % ADRN  H(1)                    00000755
        STA     0(3);                   %H(1):=H(1)+RGA                00000756
%                                                                       00000757
%  COLUMN MODIFICATION                                                  00000758
        LDL(2)  ,MASK1;                 % SET J WITH K+3 OR N           00000759
        LIT(3)  0,0,3;                                                  00000760
        CAND(2) $C1;                    % ACAR2:=0,0,K                  00000761
        CADD(2) $C3;                    % ACAR2:=0,0,K+3                00000762
        LDL(3)  $C2;                    % ACAR3:=0,0,K+3                00000763
```

```
        CSUB(2)   $C0;                  % ACAR2:=0,0,K+3-N                    00000764
        CTSBF(2)  40,NINTOJ;            % IF 40-TH BIT OF ACAR2 IS 0,J:=N      00000765
        STL(3)    ,J;                   %,J:=K+3 IF K+3<N                    00000766
        SKIP      ,CONT5;               %,J:=N IF K+3 GEQ N                  00000767
NINTOJ: STL(0)    ,J;                                                        00000768
%                                                                            00000769
    DEFINE  LRGXJ =                                                          00000770
        LDL(2)    ,L;                                                        00000771
        LDL(3)    ,J;                                                        00000772
        IXG       =0(2);                %I:=1 IF L<RGX                       00000773
        JXL       =0(3);                %J:=1 IF RGX<J                       00000774
        SETE      E OR =E;                                                   00000775
        SETE      I AND E;                                                   00000776
        SETE      J AND E;              %E:=1 IF L<RGX<J                     00000777
        IXE       =0(2);                %I:=1 IF L=RGX                       00000778
        JXF       =0(3);                %J:=1 IF RGX=J                       00000779
        SETE      I OR E;                                                    00000780
        SETE      J OR E ##;            % E:=1 IF L LEQ RGX LEQ J            00000781
%                                                                            00000782
CONT5:  SETE      E OR =F;                                                   00000783
        LDX       PEN;                  %RGX:=PEN                            00000784
        RTL       $X,0(1);              % ROUTE RGX BY K                     00000785
        LDX       $R;                                                        00000786
        LRGXJ     ;                                                          00000787
        SETH      E OR E;               % H IS SET FOR H[I,K]                00000788
        SETE1     E OR E;                                                    00000789
        LDL(2)    ,X;                                                        00000790
        LDL(3)    ,H;                                                        00000791
        LDA       *0(3);                %RGA:=H[I,K]                         00000792
        MLRN      $C2;                                                       00000793
        LDS       $A;                   % RGS:=H[I,K]*X                      00000794
        SETE      E OR =F;                                                   00000795
        SETE1     E OR E;                                                    00000796
        RTL       $X,1;                                                      00000797
        LDX       $R;                   % ROUTE RGX BY 1                     00000798
        RTL       $S,1;                                                      00000799
        LDS       $R;                   % ROUTE RGS BY 1                     00000800
        LRGXJ     ;                                                          00000801
        LDL(2)    ,Y;                                                        00000802
        SETG      E OR E;               % G IS SET FOR H[I,K+1]              00000803
        SETE1     E OR E;                                                    00000804
        LDL(3)    ,H;                                                        00000805
        LDA       *0(3);                %RGA:=H[I,K+1]                       00000806
        MLRN      $C2;                  % $A:=H[I,K+1]*Y                     00000807
        ADRN      $S;                                                        00000808
        LDS       $A;                   %$S:=X*H[I,K]+H[I,K+1]               00000809
%  COMPUTE H[I,K+2]=H[I,K+2]-(X*H[I,K]+Y*H[I,K+1]+Z*H[I,K+2])*R              00000810
%                                          IF NOTLAST=1                      00000811
        SETE      E OR =E;                                                   00000812
        SETE1     E OR E;                                                    00000813
        LDL(2)    ,NOTLAST;             % IF NOTLAST=0, GO TO CONT6          00000814
        ZERT(2)   ,CONT6;               % NO NEED TO COMPUTE H[I,K+2]        00000815
        RTL       $X,1;                                                      00000816
        LDX       $R;                                                        00000817
        RTL       $S,1;                                                      00000818
```

```
        LDS     $R;                                                     00000819
        LRGXJ   ;                                                       00000820
        SETE1   E OR E;             %E,E1:=1 FOR H[I,K+2]               00000821
        LDL(2)  ,Z;                                                     00000822
        LDL(3)  ,H;                                                     00000823
        LDA     *0(3);              %RGA:=H[I,K+2]                      00000824
        LDL(3)  ,R;                                                     00000825
        MLRN    $C2;                %RGA:=Z*H[I,K+2]                    00000826
        ADRN    $S;                 %RGS:=X*H[I,K]+Y*H[I,K+1]           00000827
        LDS     $A;                 %                 +Z*H[I,K+2]       00000828
        CHSA    ;                                                       00000829
        MLRN    $C3;                                                    00000830
        LDL(3)  ,H;                                                     00000831
        ADRN    *0(3);              % ADRN  *H                         00000832
        STA     *0(3);              % STA   *H                         00000833
        SETE    E OR -E;                                                00000834
        SETE1   E OR E;                                                 00000835
        RTL     $X,-1;                                                  00000836
        LDX     $R;                                                     00000837
        RTL     $S,-1;                                                  00000838
        LDS     $R;                                                     00000839
%  COMPUTE H[I,K+1]=H[I,K+1]-(X*H[I,K]+Y*H[I,K+1]+Z*H[I,K+2])*R         00000840
CONT6:  SETE    G AND E;                                                00000841
        SETE1   E OR E;             % E,E1:=G                          00000842
        LDL(2)  ,Q;                                                     00000843
        LDA     $S;                                                     00000844
        CHSA    ;                                                       00000845
        MLRN    $C2;                                                    00000846
        LDL(3)  ,H;                                                     00000847
        ADRN    *0(3);              % ADRN  *H                         00000848
        STA     *0(3);              % STA   *H                         00000849
%  COMPUTE H[I,K]:=[I,K]-(X*H[I,K]+Y*H[I,K+1]+Z*H[I,K+2])               00000850
        SETE    E OR -E;                                                00000851
        SETE1   E OR E;                                                 00000852
        RTL     $X,-1;                                                  00000853
        LDX     $R;                 % ROUTE RGX BY -1                  00000854
        RTL     $S,-1;              % ROUTE RGS BY -1                  00000855
        LDA     $R;                                                     00000856
        SETE    H AND E;            % E,E1:=H                          00000857
        SETE1   E OR E;                                                 00000858
        CHSA    ;                                                       00000859
        LDL(3)  ,H;                                                     00000860
         ADRN    *0(3);                 % ADRN  *H                     00000861
        STA     *0(3);              % STA  *H,RGA:=H[I,K]-RGS           00000862
CONT3:  TXLTM(1) ,NXKJ;                                                 00000863
        SKIP    ,NXNJ;                                                  00000864
NXKJ :  JUMP    NEXTK;                                                  00000865
%  END OF K-LOOP                                                        00000866
%  REARRANGE MATRIX INTO STANDARD FORM                                  00000867
NXNJ :  LIT(1)  -1,1,0;                                                 00000868
        CADD(1) $C0;                %$C1:=-1,0,N                        00000869
        SETE    E OR -E;                                                00000870
        SETE1   E OR E;                                                 00000871
DTAR2:  LDL(3)  ,H;                                                     00000872
```

```
        CADD(3)   $C1;                                                          00000873
        LDA       0(3);                 % LDA  H(1)                          00000874
        LIT(2)    =0.0.64;                                                      00000875
        CSUB(2)   $C1;                                                          00000876
        RTL       $A.0(2);                                                      00000877
        STR       0(3);                 % STR  H(1)                          00000878
        TXGTM(1)  .OTAR2;               % TEST AND MODIFY ACAR2 BY -1         00000879
        JUMP      NEXTIT;                                                       00000880
% IF ALGORITHM FAILS.G:=1 WHERE SUBDIAGONAL ELEMENTS CONVERGED                  00000881
FAIL :;SETE       E OR =F;                                                      00000882
        LDX       PEN;                                                          00000883
        IXG       =0(0);                                                        00000884
        LDL(3)    .N;                                                           00000885
        JXL       =0(3);                % JXL  =N                            00000886
        SETE      I AND F;                                                      00000887
        SETE      J AND F;                                                      00000888
        JXE       =0(3);                % JXE  =N                            00000889
        SETG      J OR E;               % ACARO CONTAINS PRESENT MATRIX      00000890
        JUMP      ETG;                  % SIZE                               00000891
%  IF PROCESS IS COMPLETED.G:=1 FOR ALL BITS                                    00000892
FIN  :;SETE       E OR =F;                                                      00000893
        LDX       PEN;                                                          00000894
        IXG       =0;                                                           00000895
        LDL(3)    .N;                                                           00000896
        JXL       =0(3);                % JXL  =N                            00000897
        SETE      I AND F;                                                      00000898
        SETE      J AND F;                                                      00000899
        IXE       =0;                                                           00000900
        JXE       =0(3);                % JXE  =N                            00000901
        SETE      I OR E;                                                       00000902
        SETG      J OR E;               %G:=1 FOR 0 LEQ RGX LEQ N             00000903
%  FIND  ONE ROOT                                                               00000904
EIG  :;SETE       E OR =F;                                                      00000905
        LDL(3)    .N;                                                           00000906
        IXL       =0(3);                % IXL  =N                            00000907
        JXE       =0(3);                % JXE  =N                            00000908
        SETE      I AND E;                                                      00000909
        SETE      J OR E;                                                       00000910
        SETE      G AND E;                                                      00000911
        SETE1     E OR E;                                                       00000912
        LDL(3)    .H;                                                           00000913
        LDA       *0(3);                % LDA  *H                            00000914
        LDL(3)    .WR;                                                          00000915
        STA       0(3);                 % STA  WR                            00000916
        CLRA      ;                                                             00000917
        LDL(3)    .WI;                                                          00000918
        STA       0(3);                 % IMAGINARY PART:= 0                  00000919
%  TWO ROOTS ARE FOUND                                                          00000920
        LDL(2)    .SOLV;                % SET APPROPRIATE BIT PATTERN         00000921
        ZERF(2)   .TWRT;                                                        00000922
        JUMP      HQREND;                                                       00000923
TWRT :  SETE      E OR =E;                                                      00000924
        IXG       =0;                                                           00000925
        LDL(3)    .N;                                                           00000926
        JXL       =0(3);                % JXL  =N                            00000927
```

```
      SETE      I AND E;                                                                    00000928
      SETE      J AND E;                                                                    00000929
      IXE       =0;                                                                         00000930
      JXE       =0(3);              % JXE  =N                                               00000931
      SETE1     J OR E;                                                                     00000932
      SETH      G AND E1;           %H:=1 IF 1 LEQ RGX LEQ N AND G=1                        00000933
      SETE1     I OR E;                                                                     00000934
      SETJ      G AND E1;       %J:=1 IF 0 LEQ RGX LEQ N-1 AND G=1                          00000935
      SETE      E OR -E;                                                                    00000936
      SETE      H AND E;                                                                    00000937
      SETG      J OR E;                % G:=1 IF 0 LEQ RGX LEQ N AND G=1                   00000938
      SETE      E OR -E;                                                                    00000939
      SETE      J AND E;            % E:=J                                                  00000940
      SETE1     E OR E;                                                                     00000941
      LDL(3)    .H;                                                                         00000942
      LDA       *1(3);              % LDA  *H+1                                             00000943
      RTL       $A,1;               % ROUTE RGA BY 1                                        00000944
      SETE      E OR -E;                                                                    00000945
      SETE      H AND E;                                                                    00000946
      SETE1     E OR E;             % E:=H                                                  00000947
      LDA       *-1(3);             % LDA  *H-1                                             00000948
      MLRN      $R;                                                                         00000949
      STA       TMP1;               % TMP1:=H[N-1,N]*H[N,N-1]                               00000950
      SETE      E OR -E;                                                                    00000951
      SETE      G AND E;            % E,E1:=G                                               00000952
      SETE1     E OR E;                                                                     00000953
      LDL(3)    .H;                                                                         00000954
      LDA       *0(3);              % LDA  *H                                               00000955
      RTL       $A,1;               % ROUTE RGA BY 1                                        00000956
      SETE      H AND E;                                                                    00000957
      SETE1     E OR E;             % E,E1:=H                                               00000958
      LDS       $R;                                                                         00000959
      ADRN      $R;                                                                         00000960
      SHAMR     1;                                                                          00000961
      NORM      ;                                                                           00000962
      STA       TMP2;               %TMP2:=(H[N,N]+H[N-1,N-1])/2                           00000963
      LDA       $S;                                                                         00000964
      LDL(3)    .H;                                                                         00000965
      SBRN      *0(3);              % SBRN  *H                                              00000966
      SHAMR     1;                                                                          00000967
      NORM      ;                                                                           00000968
      MLRN      $A;                 %RGA:=((H[N-1,N-1]-H[N,N])/2)**2                       00000969
      ADRN      TMP1;               % RGA:=RGA+H[N-1,N-1]*H[N,N]                           00000970
      ISN       ;                   %I:=SIGN OF RGA                                        00000971
      SETC(0)   I;                  % ACARO:= I BIT                                        00000972
      SAP       ;                   % RGA:= ABS(RGA)                                       00000973
      STL(0)    .SV0;               % SAVE ACARO                                           00000974
      CLC(3)    ;                                                                         00000975
      SLIT(3)   SQRT;                                                                     00000976
      EXCHL(3)  $ICR;               % RGA:=SQRT(ABS(RGA))                                  00000977
      STA       TMP3;               % TMP3:=RGA                                            00000978
      LDL(0)    .SV0;               % RESTORE ACARO                                        00000979
%  FIND COMPLEX PAIR                                                                      00000980
      SETC(1)   H;                  % ACAR1,ACAR2:=H BIT                                   00000981
```

```
        LDL(2)    $C1;
        CAND(1)   $C0;                % ACAR1:=H BIT AND SIGN
        CAND(1)   .SOLV;              % ACAR1:=ACAR1 AND SLTN PATTERN
        ZERT(1)   .FREAL;
        LDEE1     $C1;
        LDL(3)    .WI;
        STA       0(3);               % STA  WI
        LDS       TMP2;               % FOUND WR[N] AND WI[N]
        LDL(3)    .WR;
        STS       0(3);               % STA  WR
        CSHL(1)   1;                  % SHIFT ACAR1 LEFT BY 1
        LDEE1     $C1;                % E,E1:=ACAR1
        RTL       $A,-1;              % ROUTE RGA BY -1
        LDA       $R;
        RTL       $S,-1;              % ROUTE RGS BY -1
        CHSA      ;
        LDL(3)    .WI;
        STA       0(3);               % STA  WI
        LDL(3)    .WR;
        STR       0(3);               % STR  WR
%  FIND REAL PAIR
FREAL;  COMPC(0)  ;                   % ACAR0:=COMPLEMENT OF ACAR0
        CAND(2)   $C0;                % ACAR2:=H AND C(SIGN)
        CAND(2)   .SOLV;
        ZERT(2)   .HQREND;
        LDEE1     $C2;
        LDA       TMP2;
        LDS       $A;
        ADRN      TMP3;
        LDL(3)    .WR;
        STA       0(3);               % STA  WR
        CLRA      ;
        LDL(3)    .WI;
        STA       0(3);               % STA  WI
        LDA       TMP3;
        CHSA      ;
        ADRN      $S;                 % RGA:=TMP2-TMP3
        CSHL(2)   1;
        LDEE1     $C2;
        RTL       $A,-1;
        LDL(3)    .WR;
        STR       0(3);               % STR  WR
        CLRA      ;
        LDL(3)    .WI;
        STA       0(3);               % STA WI
HQREND:;
        LDL(0)    .HQSVE;
        LDEE1     $C0;                % RESTORE E,E1
        LDL(0)    .HQSV0;
        LDL(1)    .HQSV1;
        LDL(3)    .HQSV3;
        EXCHL(3)  $ICR;
        END       HQR.
```

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE

THE QR-ALGORITHM

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Research Report

5. AUTHOR(S) *(First name, middle initial, last name)*

Masako Ogura

6. REPORT DATE
September 1, 1971

7a. TOTAL NO. OF PAGES
44

7b. NO. OF REFS

8a. CONTRACT OR GRANT NO.
USAF 30-(602)-4144

b. PROJECT NO.
ARPA Order 788

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

CAC Document No. 12

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

Copies may be requested from the address given in (1) above.

11. SUPPLEMENTARY NOTES
None

12. SPONSORING MILITARY ACTIVITY
Rome Air Development Center
Griffiss Air Force Base
Rome, New York 13440

13. ABSTRACT

The implementation of QR-algorithm on ILLIAC IV is described. An ASK subroutine for computing all eigenvalues of a real Hessenberg matrix of order less than or equal to 64 by this algorithm is attached. The QR-transformation consists of the decomposition of the matrix A_k into the product of a unitary matrix Q_k and an upper triangular matrix R_k, and forming A_{k+1} by post-multiplying R_k by Q_k, where $A_1 = A$ is the original matrix. All eigenvalues are either isolated on the diagonal or are eigenvalues of a 2 x 2 diagonal submatrix as $k \to \infty$.

DD FORM 1 NOV 65 1473

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Matrix Algebra